2.23 MOVING TARGET INDICATOR

The doppler frequency shift produced by a moving target may be used in a pulse radar to determine the relative velocity of a target or, more importantly, to discriminate *desired* moving targets from *undesired* stationary objects such as ground clutter. A pulse radar that uses doppler frequency shift to discern moving from fixed targets is called a moving target indicator (MTI) or a pulse doppler radar. The distinction between them is based on the sampling rate, chosen to avoid ambiguities in either range (time delay) or doppler frequency (relative velocity) measurements. MTI usually refers to a radar in which the Pulse Repetition Frequency (PRF) is chosen low enough to avoid range ambiguities, but with the consequence of ambiguous frequency measurement which results in blind speeds. Pulse doppler radars, on the other hand, typically use high PRFs thus avoiding blind speeds at the expense of ambiguous range measurement. Radar systems employing multiple or staggered PRFs to avoid blind speeds are usually classed as MTI if the average PRF would cause blind speeds.

Although MTI is a general term that applies to any pulse radar system that extracts and makes use of doppler information, it is usually reserved for those systems characterized by phase detectors using delay-line cancellation. The general MTI methodology begins with the transmission of a coherent pulse train, where the term coherent implies a constant RF phase relationship from pulse-to-pulse. On reception, video signals from echo returns are divided into two channels, a direct line and a second channel which passes through a time-delay device (t = 1/PRF). The output of the delayed channel is then subtracted from the output of the direct channel. A block diagram of a single delay-line canceler is shown in Figure 2.23-1.

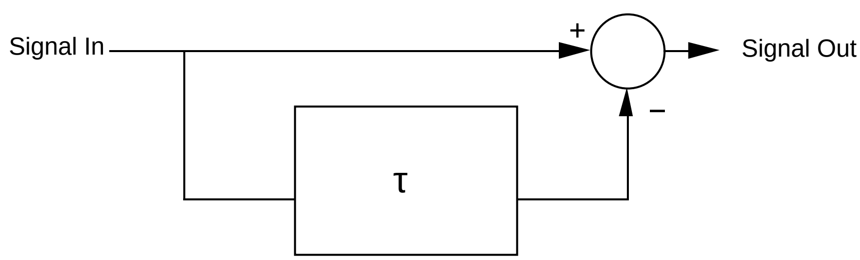

FIGURE 2.23-1. Block Diagram of a Single Delay-Line Canceler.

Because the transmitted pulse train is coherent, subtraction of the delayed pulses will completely cancel in the case of stationary scattering sources whereas pulse echoes with superimposed doppler shift due to target motion will not completely cancel from pulse to pulse. The resulting signal output fluctuates at the doppler frequency of the moving target; thus, the delay-line canceler acts as a bandstop filter that heavily attenuates the direct current (DC) component of the return signal and passes the alternating current (AC) components. Because clutter is fundamentally DC in nature, delay-line cancellation provides a means of clutter rejection even when clutter returns are two or three orders of magnitude greater than target returns.

The MTI functional element is intended to simulate a radar's extraction of the doppler frequency shift that enables it to distinguish the signal returns of moving targets from those of stationary scattering sources such as ground clutter.

2.23.1 Functional Element Design Requirements

1. The MTI functional element will simulate the effect of two-stage delay-line cancellation MTI processing at time-steps equal to one or more pulse repetition intervals.

   The effect of two-stage delay-line processing can be expressed deterministically as a frequency response. To accommodate PRI and multiple PRI time-step intervals, *RADGUNS* will use the magnitude of the delay-line frequency responses as an attenuation factor for MTI processing.

2. The MTI functional element will simulate the effect of delay-line cancellation MTI processing on an eleven-pulse staggered waveform.

   The use of staggered PRF waveforms reduce the effects of frequency domain nulls in the MTI response caused by PRF-induced blind speeds. *RADGUNS* will incorporate an eleven-pulse staggered waveform into the MTI signal attenuation factor.

3. The MTI functional element will use three operation modes: MTI processing on, MTI processing off, and a dynamic MTI processing mode.

   RADGUNS will allow the user to select whether MTI processing will be constantly active, constantly inactive or active only when tracking targets below a pre-specified elevation.

4. The MTI functional element will process target, clutter, noise and ECM signals while in tracking mode. It will process only target and clutter signals while in acquisition mode.

2.23.2 Functional Element Design Approach

The normalized response of a double delay-line canceler with *N* interpulse periods is given as (Reference 23):

$$|H(\omega)|^2 = 1 - \frac{2}{3(N-1)}\left(\cos\omega T_1 + \cos\omega T_N\right) - \frac{4}{3(N-1)}\sum_{n=2}^{N-1}\cos\omega T_n + \frac{1}{3(N-1)}\sum_{n=1}^{N-1}\cos\left[\omega\left(T_n + T_{n+1}\right)\right] \qquad [2.23\text{-}1]$$

where:

ω	=	target return doppler frequency (rad/sec)
N	=	number of interpulse periods
T	=	pulse repetition interval (sec)

The doppler frequency of the target is given by:

$$\omega = \frac{2V_r}{\lambda} \qquad [2.23\text{-}2]$$

where:

V_r = target's radial velocity (m/s)

λ = radar's wavelength (m)

The target's doppler angular frequency in radians per second is given by:

$$\begin{aligned}\Omega &= 2\pi\omega \\ &= \frac{4\pi}{\lambda} V_r\end{aligned} \qquad [2.23\text{-}3]$$

RADGUNS has the capability to model AAA systems in motion. Because uniform clutter is assumed, the radial velocity of Equation [2.23-3] is replaced with the weapon's speed in the calculation of the apparent doppler frequency of clutter, or:

$$\Omega_{cl} = \frac{4\pi}{\lambda}\sqrt{V_x^2 + V_y^2 + V_z^2} \qquad [2.23\text{-}4]$$

where:

V_x = weapon velocity in x direction

V_y = weapon velocity in y direction

V_z = weapon velocity in z direction

The doppler frequency of all jammers is set to the doppler frequency of the target.

Design Element 23-1: Double Delay-Line Staggered Response

RADGUNS will simulate an eleven-pulse staggered PRF system using Equation [2.23-1]. Setting *N* equal to 11 gives:

$$|H(\omega)|^2 = 1 - \frac{2}{30}\left(\cos\omega T_1 + \cos\omega T_{11}\right) - \frac{4}{30}\sum_{n=2}^{10}\cos\omega T_n + \frac{1}{30}\sum_{n=1}^{10}\cos\left[\omega\left(T_n + T_{n+1}\right)\right] \qquad [2.23\text{-}4]$$

Design Element 23-2: Staggered Pulse Repetition Frequency

The PRF values used in *RADGUNS* were obtained from exploitation data.

Design Element 23-3: Response Limiting

RADGUNS will limit the MTI attenuation factor to a minimum value that corresponds to intelligence data.

Design Element 23-4: MTI Mode Selection

RADGUNS will determine whether or not to perform MTI processing based on user input and, possibly, target location. The user will be able to select any of three options for MTI processing: on, off, or dynamic. If the user selects on or off, the MTI switch will be set accordingly. If the user selects dynamic, the MTI switch will be turned on when the target is below an elevation angle determined by the radar's beamwidth. There is a delay associated with switching between modes.

2.23.3 Functional Element Software Design

This section contains the software design necessary to implement the functional element requirements and the design approach outlined above. It is organized as follows: the first subsection describes the subroutine hierarchy and gives descriptions of the relevant subroutines; the next subsection contains a logical flow chart and describes all important operations represented by each block in the chart; the last subsection contains a description of all input and output data for the functional element as a whole and for each subroutine which implements MTI.

MTI Subroutine Design. Figure 2.23-2 shows the calling sequence of the MTI functional element within the entire model structure. Functions which implement the MTI functional element appear in shaded blocks. Each of these subroutines is briefly described in Table 2.23-1. Subroutines that directly implement the MTI functional element appear in shaded blocks.

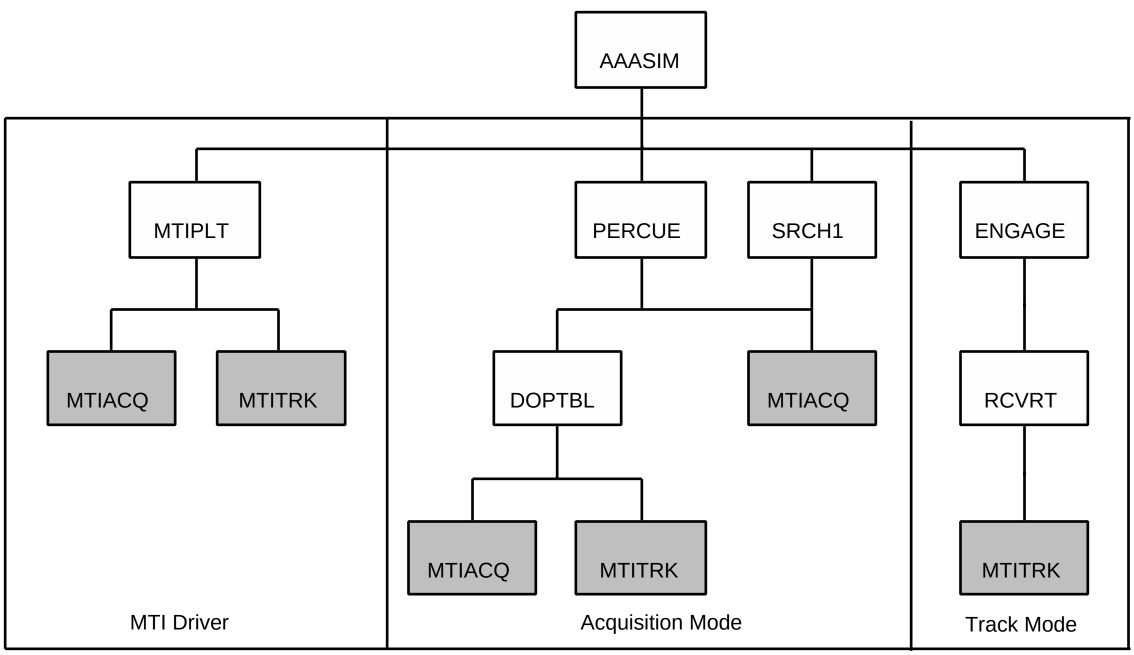

FIGURE 2.23-2. Subroutine Calling Tree.

TABLE 2.23-1. Subroutine Descriptions.

Name	Description
AAASIM	Main routine to simulate AAA system
ENGAGE	Controls the system while tracking and engaging targets
DOPTBL	Calculates the MTI response to a rotating blade
MTIACQ	Calculates the acquisition radar MTI attenuation factor as a function of doppler angular frequency
MTIPLT	Outputs MTI response as a function of doppler angular frequency to unit 15
MTITRK	Calculates the track radar MTI attenuation factor as a function of doppler angular frequency
PERCUE	Searches for target with antenna perfectly cued to target position
RCVRT	Radar receiver model that processes target, clutter, and jammer returns
SRCH1	Search for target in sector search or circular scan mode

Functional Flow Diagram for Acquisition Mode. The MTIACQ function is used to calculate attenuation factors for signals being processed by the receiver in acquisition mode. Target and clutter signals are attenuated using calls to function MTIACQ in subroutines PERCUE and SRCH1. These routines are used to simulate the various search modes that can be employed with the acquisition radar.

The functional flow diagram of Figure 2.23-4 shows the subroutines that call and are directly affected by function MTIACQ. Subroutine names are enclosed in parentheses and appear in bold face. The numbered blocks are discussed below. Subroutines DOPTBL and MTIACQ are discussed in detail below. Variable names are italicized in the following discussion.

Block 1. Several variables that affect the MTI function are initialized. *MTIDEL*, the minimum time between MTI state changes is set to 5 seconds. *MTISW*, the MTI switch character flag is filled with blank spaces. *MTITIM*, the next time that an MTI state change can occur is set to 0 seconds.

Blocks 2 and 3 are executed if the RCS file contains blade information.

Block 2. Subroutine DOPTBL is called to calculate the MTI response to the helicopter blade.

Block 3. Subroutine PULSES is called to determine the number of pulses between blade flashes.

Blocks 4 through 14 are executed until the a simulation termination condition exists or the target is detected.

Block 4. Time is incremented (by 0.1 seconds in perfect cueing search mode and by the scan rate for circular/sector search modes) and the target is moved to its new position.

Block 5. If a termination condition exists (determined by function ENDRUN), control returns to the main program, AAASIM.

Block 6. If the current time is greater than the next possible MTI state change time, the MTI switch is initialized. If the user has selected the dynamic MTI mode (common variable *MTICHG* is true) and the target elevation angle is below the elevation angle at which the MTI switches, the MTI switch is set to 'ON' (*MTISW*). If the switch has changed state, subroutine EVENT writes a message to the .EVT file. The earliest time of the next state change is set to the current time plus the delay time. If the user has selected the dynamic MTI mode but the target is not below the MTI switch elevation angle, the switch is set to 'OFF'. Subroutine EVENT writes a message to the .EVT file and the earliest time of the next state change is set to the current time plus the delay time. Otherwise, subroutine EVENT is called to write the user selected position of the MTI switch to the .EVT file (either ON or OFF). The time of the next state change is set to 9999.9 indicating a permanent switch position.

Blocks 7 through 11 are executed if the MTI switch is in the 'ON' position.

Block 7. The target doppler frequency is calculated using Equation [2.23-3].

Block 8. The body return is calculated via the radar range equation (function RDREQA). This value is multiplied by the MTI attenuation factor for the doppler frequency of interest calculated in function MTIACQ.

Block 9. If the RCS file contains blade information, the blade return is calculated via the radar range equation (function RDREQA) modified by the MTI attenuation factor for the blade (contained in array *DOPACQ*). The target signal is calculated as the sum of the body return and the blade return.

Block 10. The doppler frequency of the clutter as seen by the threat is calculated by Equation [2.23-4].

Block 11. Function CLUTG calculates the signal returned from a ground clutter patch. This value is modified by the MTI attenuation factor for the clutter doppler frequency. Execution resumes following Block 14.

Block 12. If the MTI switch is in the 'OFF' position, function RDREQA calculates the target return.

Block 13. If the RCS file contains blade information, the blade return is calculated via function RDREQA. The target return is calculated as the sum of the blade return and the body return.

Block 14. The clutter return is calculated via function CLUTG.

If the user has selected the probability of detection model, subroutine PDET is used to determine the probability of detection for a given probability of false alarm and Swerling case. If the probability of detection exceeds the user specified detection threshold, the target is detected, and subroutine AAASIM calls subroutine ENGAGE to track and engage the target. Otherwise, Blocks 4 through 14 are repeated.

If the user has selected the threshold detection model, function DETECT is used to determine if the signal-to-noise ratio exceeds the specified detection threshold. If so, the target is detected, and subroutine AAASIM calls subroutine ENGAGE to track and engage the target. Otherwise, Blocks 4 through 14 are repeated.

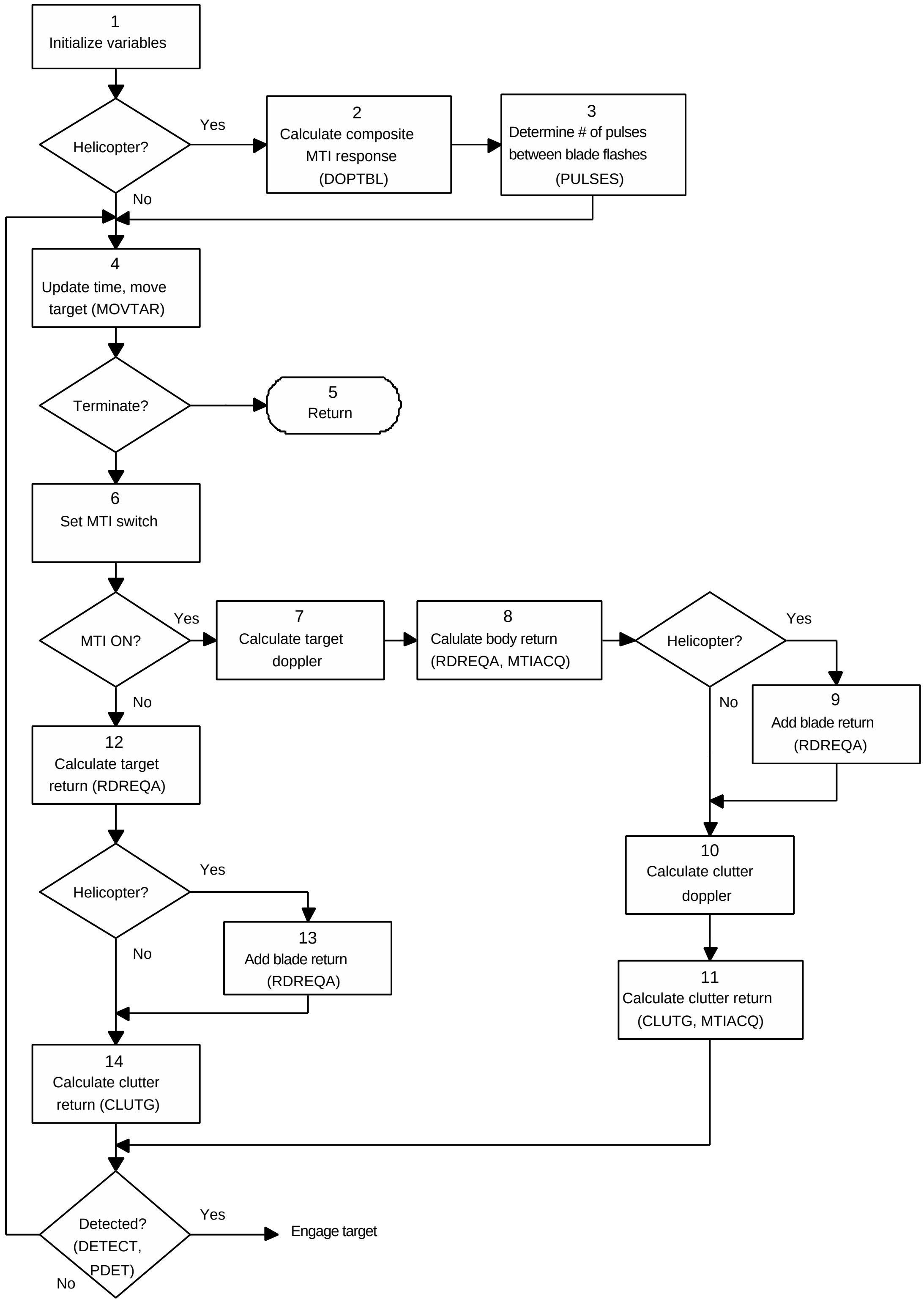

FIGURE 2.23-3. Acquisition Mode Functional Flow Diagram.

<u>Subroutine DOPTBL</u>. Subroutine DOPTBL generates the MTI response to a rotating helicopter blade based on the doppler shift from the combination of the blade motion and the radial velocity of the target. Each blade is divided into 32 segments. The overall response is the average of the individual segment responses. The MTI response of the acquisition and track radars are stored in 2 x 401 arrays *DOPACQ* and *DOPTRK*, respectively. The first row holds the response to a target and blade moving in the same direction; the second row contains the response to a target and blade moving in opposite directions. The response is calculated for radial velocities ranging from 0 to 400 m/s in 1 m/s increments. A logic flow chart is shown in Figure 2.23-5 and discussed below.

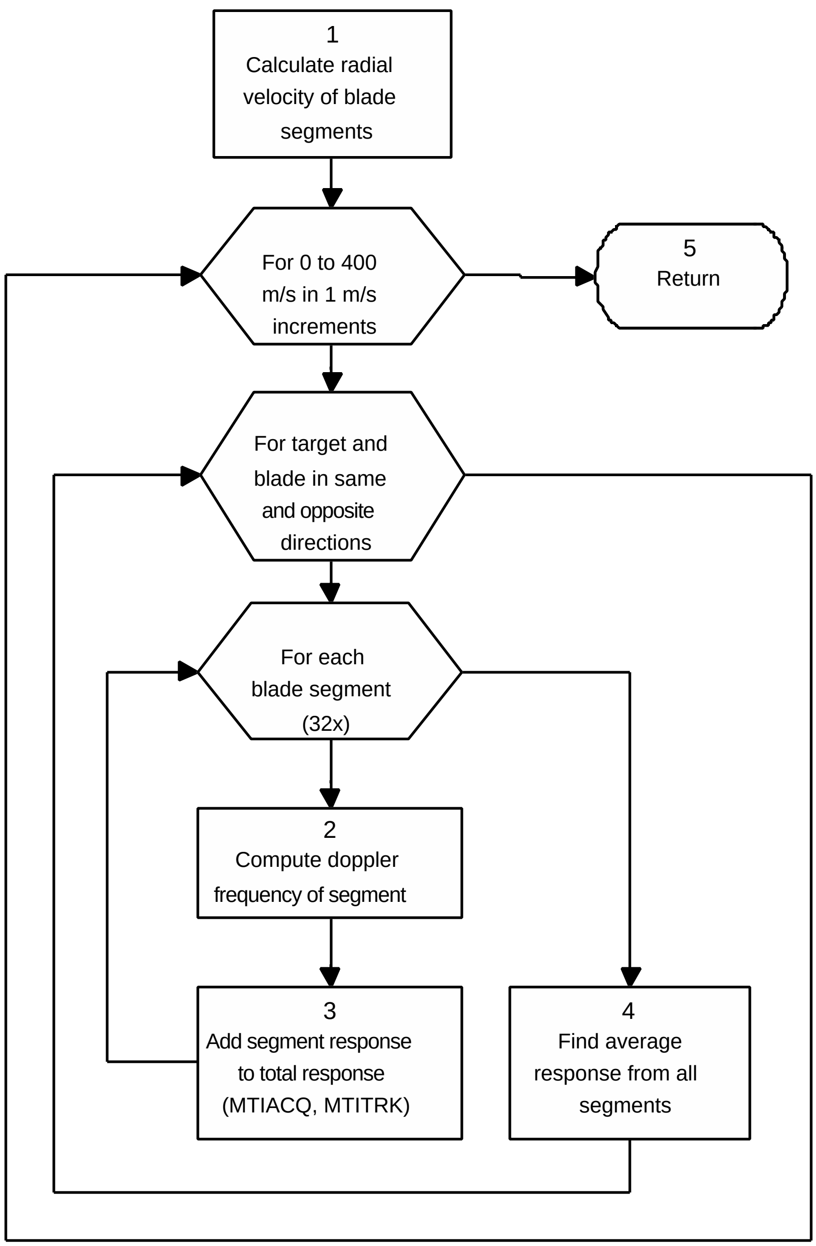

FIGURE 2.23-4. DOPTBL Logic Flow Chart.

Block 1. The radial velocity of each blade segment is calculated from the blade's length and speed of rotation.

Subroutine DOPTBL consists of a series of nested loops. The outer loop is executed 401 times (target radial velocity is varied from 0 to 400 m/s in increments of 1 m/s). The middle loop is executed twice. For each radial velocity increment, the MTI response is calculated for a target and blade moving in the same and opposite directions. The inner loop is executed 32 times (once for each blade segment).

Block 2. The doppler frequency of the blade for both the acquisition and track radars is computed. If the inner loop has not been executed for the current target radial velocity, the blade doppler is computed as the sum of the target doppler and the blade doppler (target and blade moving in same direction); otherwise, the doppler frequency of the blade is computed as the difference between the blade and target doppler frequencies (target and blade moving in opposite directions).

Block 3. Functions MTIACQ and MTITRK are used to calculate the acquisition and track radar MTI attenuation factors respectively. The MTI factors for each segment are summed.

Block 4. Once all 32 factors have been summed, an average response to the current target doppler increment is calculated and stored in variables *DOPACQ* and *DOPTRK*. If blocks 2 through 4 have not been executed previously for the current target radial velocity, the middle loop variable is incremented and execution returns to block 2. Otherwise, the target radial velocity is incremented and execution returns to the outer loop.

Block 5. Once the MTI factors for both the acquisition and track radars for target radial velocities ranging from 0 to 400 m/s have been calculated and stored, execution returns to the calling routine.

Subroutine MTIACQ. Subroutine MTIACQ calculates the attenuation factors for signals being processed by the receiver in acquisition mode. A logic flow chart is shown in Figure 2.23-5 and discussed below.

The radial velocity of the target (*OMEGAD*) is passed as an argument to subroutine MTIACQ in addition to common variable *CLATA*, the radar clutter attenuation power ratio.

Block 1. A DATA statement is used to define array *PRI*. Each element holds a stagger period as defined by intelligence data.

Block 2. The first term of Equation [2.23-4] is calculated (*TERM1*).

Block 3. Variable *TERM2* is set to zero.

The second term of Equation [2.23-4] is calculated by executing Block 4 nine times and doing the multiplication of Block 5.

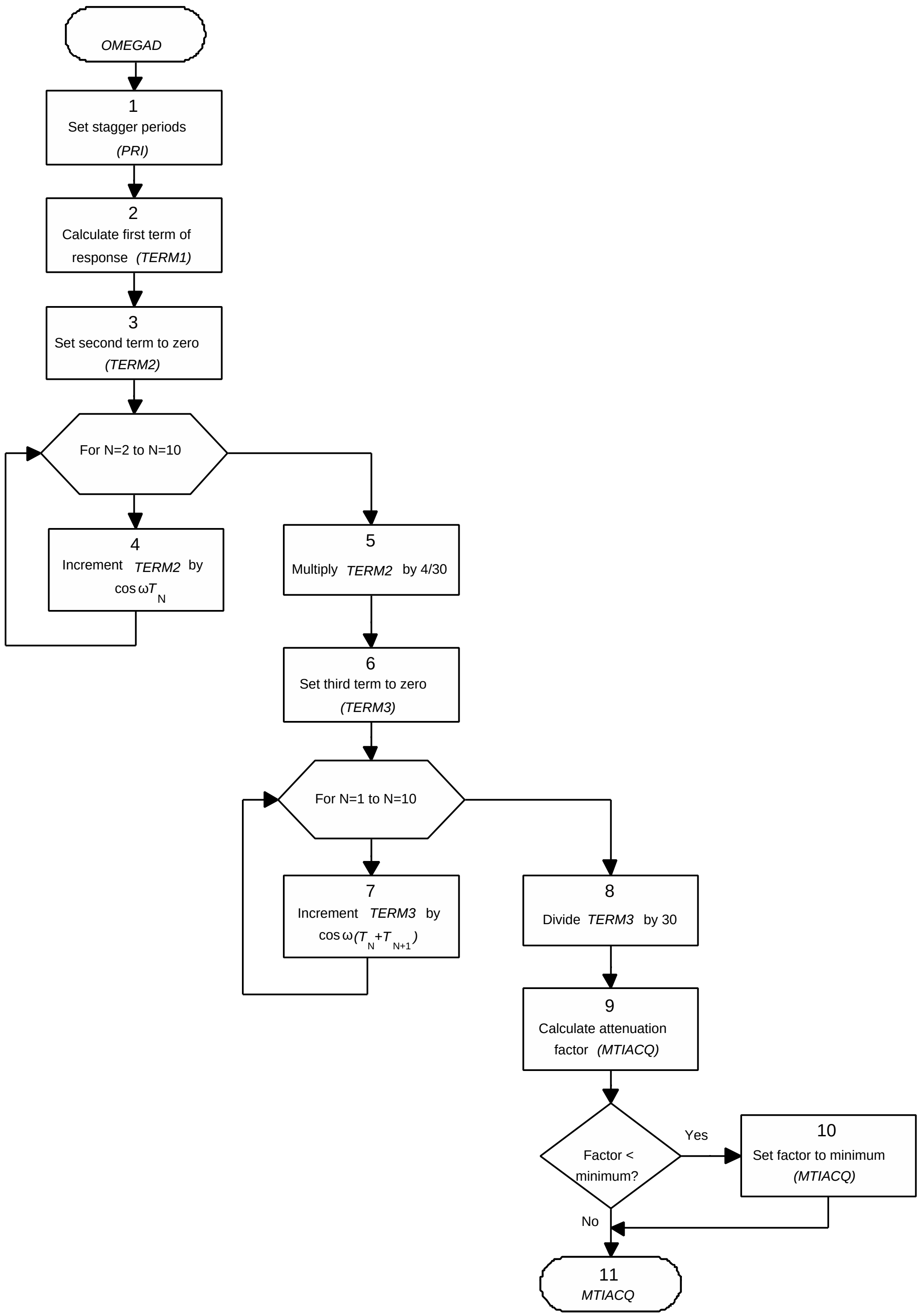

FIGURE 2.23-5. Subroutine MTIACQ Logic Flow Chart.

Block 4. For *PRI* elements 2 through 10, *TERM2* is incremented by the cosine of the product of the radial velocity of the target and the current stagger period.

Block 5. The summation is multiplied by factor of 4/30.

Block 6. Variable *TERM3* is set to zero.

The third term of Equation [2.23-4] is calculated by executing Block 7 ten times and doing the division of Block 8.

Block 7. For *PRI* elements 1 through 10, *TERM3* is incremented by the cosine of the product of the radial velocity of the target and the summation of the current and next stagger periods.

Block 8. The summation is divided by 30.

Block 9. The terms calculated in blocks 2, 5, and 8 are combined using Equation [2.23-4] to form the MTI attenuation factor (*MTIACQ*).

Block 10. If the resulting factor is less than the minimum clutter power ratio defined by intelligence data (*CLATA*), the factor is set to the minimum.

Block 11. Function *MTIACQ* returns the MTI attenuation factor.

Functional Flow Diagram for Tracking Mode. The MTITRK function will be used to calculate attenuation factors for signals being processed by the receiver in track mode. During autotrack, MTI attenuation will be applied to the range channel only for the system of interest.

The functional flow diagram of Figure 2.23-7 shows the subroutines that call and are directly affected by function MTITRK. Subroutine names are enclosed in parentheses and appear in bold face. The numbered blocks are referenced in the discussion below.

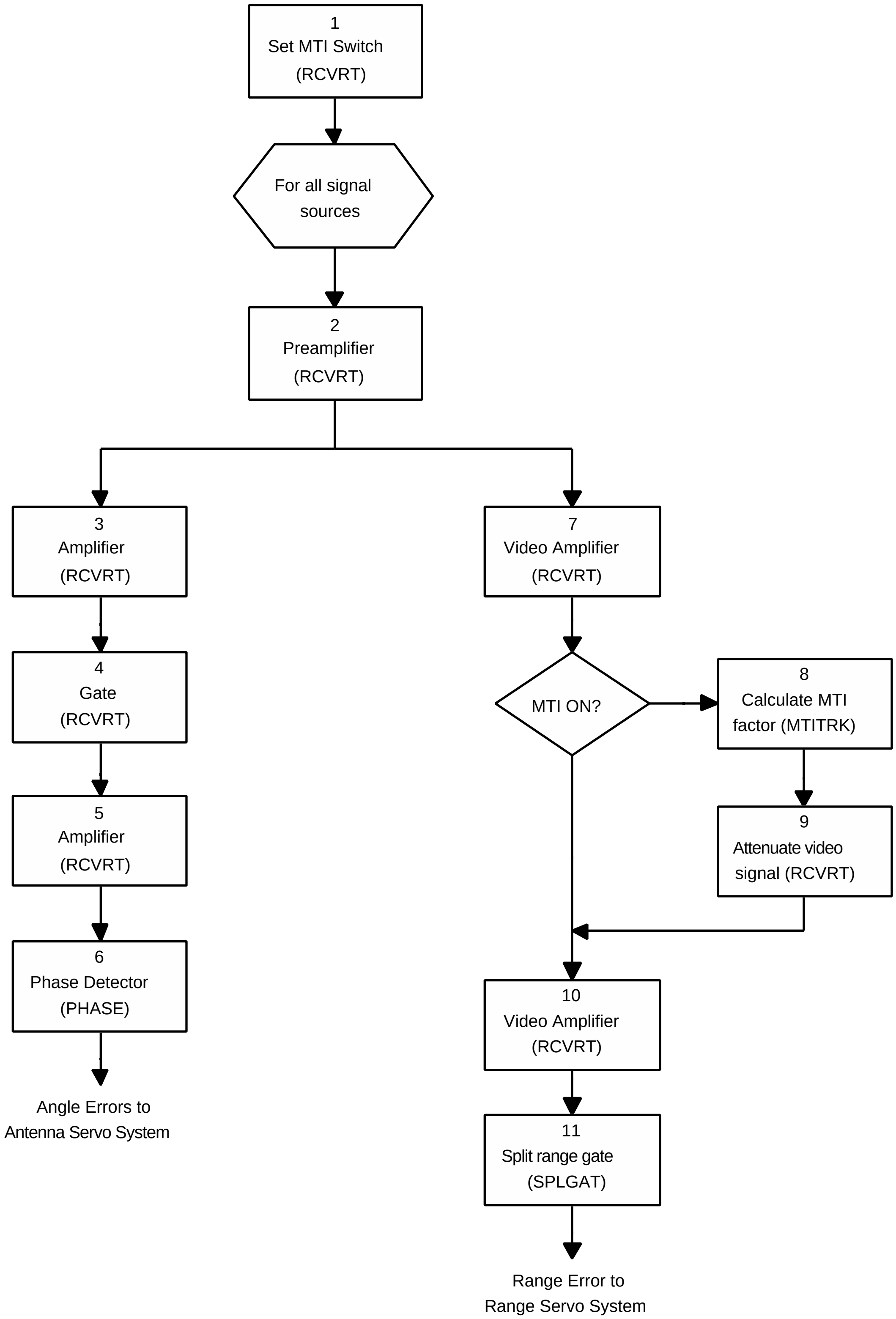

FIGURE 2.23-6. Track Mode Functional Flow Diagram.

Subroutine MTITRK. Subroutine MTITRK calculates the attenuation factors for signals being processed by the receiver in track mode in a similar manner as subroutine MTIACQ, the only differences being variable name changes. *MTIACQ* is replaced by *MTITRK*, and *CLATA* is replaced by *CLATT*.

MTI Inputs and Outputs. User inputs that affect the MTI functional element are given in Table 2.23-2 below. The outputs from this functional element are the MTI attenuation factors MTIACQ for acquisition mode and MTITRK for the range channel of tracking mode.

TABLE 2.23-2. User Inputs for MTI.

Variable Name	Description
MTISW	MTI switch; values are ON, OFF, or DYNAMIC

Note: PRF values are fixed constants in MTIACQ and MTITRK

Inputs and outputs for the two function routines that directly implement the MTI functional element are given in Tables 2.23-3 and 2.23-4.

TABLE 2.23-3. Function MTIACQ Inputs and Outputs.

FUNCTION: MTIACQ					
Inputs			**Outputs**		
Name	**Type**	**Description**	**Name**	**Type**	**Description**
CLATA	Common CLATA	Minimum value of MTI attenuation factor for acquisition	MTIACQ	Function value returned	MTI attenuation factor for acquisition
OMEGAD	Argument	Doppler angular frequency			

TABLE 2.23-4. Function MTITRK Inputs and Outputs.

FUNCTION: MTITRK					
Inputs			**Outputs**		
Name	**Type**	**Description**	**Name**	**Type**	**Description**
CLATT	Common CLATT	Minimum value of MTI attenuation factor for tracking	MTITRK	Function value returned	MTI attenuation factor for range channel during tracking
OMEGAD	Argument	Doppler angular frequency			

2.23.4 Assumptions and Limitations

All target and clutter returns are expressed in Watts. Thus, the attenuation factor which operates on these signals must be expressed as a power ratio.

The MTI attenuation factors output from functions MTIACQ and MTITRK are limited by constants CLATA and CLATT as mentioned above. These constants are system specific and were derived from intelligence data.